Lớp: IT003.O21.CTTN

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Thời gian thực hiện: 27/02 – 11/03/2023

Sinh viên thực hiện: Ngô Trần Anh Ninh (MSSV: 23521134)

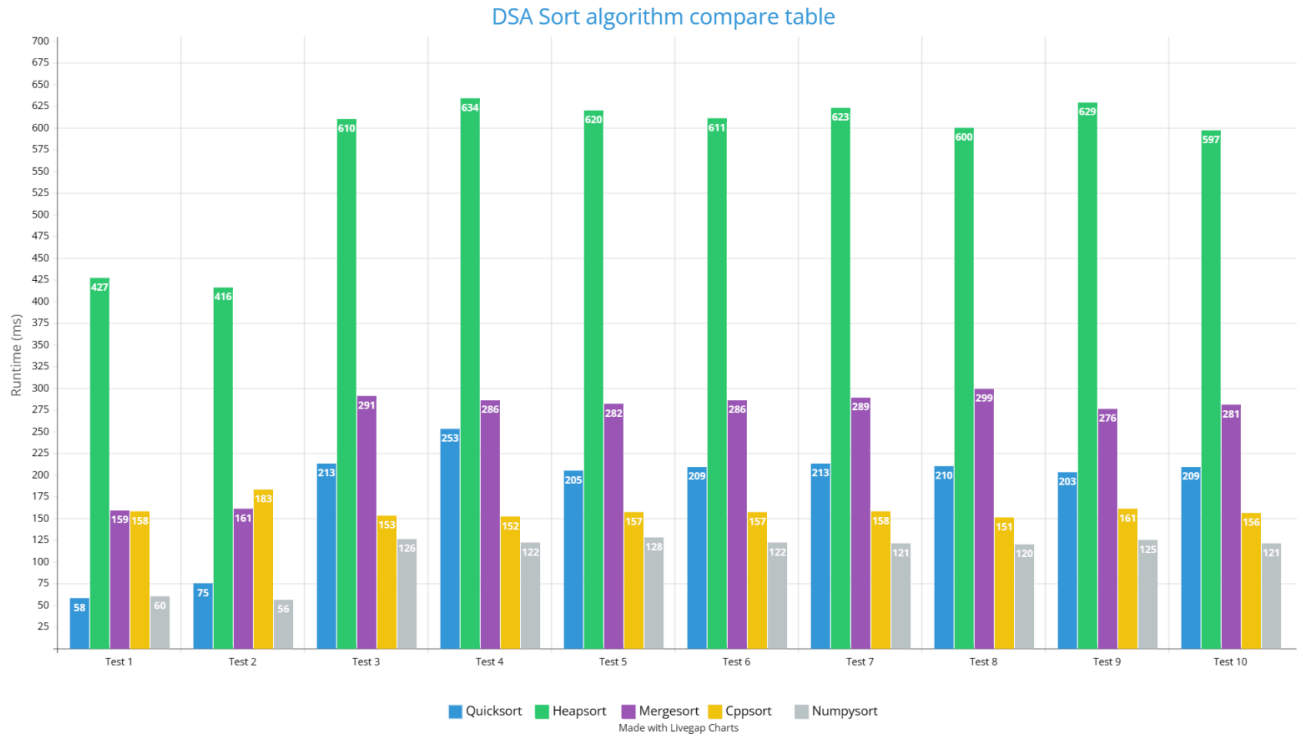
Nội dung báo cáo:

I. Kết quả thử nghiệm

1. Bảng thời gian thực hiện[1]

Dữ liệu	Thời gian thực hiện (ms)				
	Quicksort	Heapsort	Mergesort	sort (C++)	sort (numpy)
1	58	427	159	158	60
2	75	416	161	183	56
3	213	610	291	153	126
4	253	634	286	152	122
5	205	620	282	157	128
6	209	611	286	157	122
7	213	623	289	158	121
8	210	600	299	151	120
9	203	629	276	161	125
10	209	597	281	156	121
Trung bình	184.8	576.7	261	158.6	110.1

2. Biểu đồ (cột) thời gian thực hiện:



1

II. Kết luận:

- Đối với hàm sort tự cài đặt, quicksort là thuật sort đơn giản nhất và có hiệu quả tốt nhất với trung bình 184.6ms thời gian chạy trong tất cả các trường hợp
- Trong đó đặc biệt chỉ có quicksort sẽ nhanh hơn trong các trường hợp thuận lợi như mảng đã được sort sẵn (tăng - giảm dần)
- Tuy nhiên tốc độ của các hàm sort tự cài đặt chưa thể nhanh bằng các hàm sort có sẵn
- Đối với các hàm sort có sẵn là cpp sort và numpy sort:
    + Cpp sort không thể hiện sự nổi trội đối với các trường hợp đặc biệt đã sort sẵn ở test 1 và test 2 như quicksort, nhưng chạy ổn định ở hầu hết các test
    + Numpy sort dù tích hợp trên Python, chậm hơn Cpp nhiều nhưng lại có tốc độ chạy nhanh nhất, đặc biệt có tốc độ nhanh hơn hẳn trong các trường hợp thuận lợi ở test 1, 2

- C++: Vẫn nên dùng sort có sẵn vì sự ổn định, tuy nhiên trong một số trường hợp thì vẫn có thể sử dụng quick sort vẫn cho hiệu quả tốt
- Python: Numpy sort nhanh, ổn định

III. Thông tin chi tiết – link github:

vilesport/DSA-Report (github.com)